



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f583-imr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f583-imr</a>

# C8051F58x/F59x

---

12.2.1.2. Bit Addressable Locations .....	105
12.2.1.3. Stack .....	105
<b>13. Special Function Registers.....</b>	<b>106</b>
13.1. SFR Paging .....	106
13.2. Interrupts and SFR Paging .....	106
13.3. SFR Page Stack Example .....	107
<b>14. Interrupts .....</b>	<b>126</b>
14.1. MCU Interrupt Sources and Vectors.....	126
14.1.1. Interrupt Priorities.....	127
14.1.2. Interrupt Latency .....	127
14.2. Interrupt Register Descriptions .....	129
14.3. External Interrupts INT0 and INT1.....	136
<b>15. Flash Memory .....</b>	<b>138</b>
15.1. Programming The Flash Memory .....	138
15.1.1. Flash Lock and Key Functions .....	138
15.1.2. Flash Erase Procedure .....	138
15.1.3. Flash Write Procedure .....	139
15.1.4. Flash Write Optimization .....	139
15.2. Non-volatile Data Storage .....	140
15.3. Security Options .....	140
15.4. Flash Write and Erase Guidelines.....	142
15.4.1. V <sub>DD</sub> Maintenance and the V <sub>DD</sub> monitor .....	142
15.4.2. PSWE Maintenance .....	142
15.4.3. System Clock .....	143
<b>16. Power Management Modes.....</b>	<b>147</b>
16.1. Idle Mode.....	147
16.2. Stop Mode .....	148
16.3. Suspend Mode .....	148
<b>17. Reset Sources .....</b>	<b>150</b>
17.1. Power-On Reset.....	151
17.2. Power-Fail Reset/VDD Monitor .....	152
17.3. External Reset.....	153
17.4. Missing Clock Detector Reset .....	153
17.5. Comparator0 Reset .....	154
17.6. PCA Watchdog Timer Reset .....	154
17.7. Flash Error Reset .....	154
17.8. Software Reset .....	154
<b>18. External Data Memory Interface and On-Chip XRAM .....</b>	<b>156</b>
18.1. Accessing XRAM.....	156
18.1.1. 16-Bit MOVX Example .....	156
18.1.2. 8-Bit MOVX Example .....	156
18.2. Configuring the External Memory Interface .....	157
18.3. Port Configuration.....	157
18.4. Multiplexed and Non-multiplexed Selection.....	162
18.4.1. Multiplexed Configuration.....	162

---

## List of Tables

Table 2.1. Product Selection Guide .....	23
Table 3.1. Pin Definitions for the C8051F58x/F59x .....	24
Table 4.1. QFP-48 Package Dimensions .....	32
Table 4.2. QFP-48 Landing Diagram Dimensions .....	33
Table 4.3. QFN-48 Package Dimensions .....	34
Table 4.4. QFN-48 Landing Diagram Dimensions .....	35
Table 4.5. QFN-40 Package Dimensions .....	36
Table 4.6. QFN-40 Landing Diagram Dimensions .....	37
Table 4.7. QFP-32 Package Dimensions .....	38
Table 4.8. QFP-32 Landing Diagram Dimensions .....	39
Table 4.9. QFN-32 Package Dimensions .....	40
Table 4.10. QFN-32 Landing Diagram Dimensions .....	41
Table 5.1. Absolute Maximum Ratings .....	42
Table 5.2. Global Electrical Characteristics .....	43
Table 5.3. Port I/O DC Electrical Characteristics .....	47
Table 5.4. Reset Electrical Characteristics .....	48
Table 5.5. Flash Electrical Characteristics .....	48
Table 5.6. Internal High-Frequency Oscillator Electrical Characteristics .....	49
Table 5.7. Clock Multiplier Electrical Specifications .....	50
Table 5.8. Voltage Regulator Electrical Characteristics .....	50
Table 5.9. ADC0 Electrical Characteristics .....	51
Table 5.10. Temperature Sensor Electrical Characteristics .....	52
Table 5.11. Voltage Reference Electrical Characteristics .....	52
Table 5.12. Comparator 0, 1 and 2 Electrical Characteristics .....	53
Table 11.1. CIP-51 Instruction Set Summary (Prefetch-Enabled) .....	94
Table 13.1. Special Function Register (SFR) Memory Map for Pages 0x00, 0x10, and 0x0F .....	117
Table 13.2. Special Function Register (SFR) Memory Map for Page 0x0C .....	119
Table 13.3. Special Function Registers .....	120
Table 14.1. Interrupt Summary .....	128
Table 15.1. Flash Security Summary .....	141
Table 18.1. EMIF Pinout (C8051F580/1/4/5) .....	158
Table 18.2. EMIF Pinout (C8051F588/9-F590/1) .....	159
Table 18.3. AC Parameters for External Memory Interface .....	173
Table 20.1. Port I/O Assignment for Analog Functions .....	189
Table 20.2. Port I/O Assignment for Digital Functions .....	189
Table 20.3. Port I/O Assignment for External Digital Event Capture Functions ....	190
Table 21.1. Baud Rate Calculation Variable Ranges .....	213
Table 21.2. Manual Baud Rate Parameters Examples .....	215
Table 21.3. Autobaud Parameters Examples .....	216
Table 21.4. LIN Registers* (Indirectly Addressable) .....	221
Table 22.1. Background System Information .....	231
Table 22.2. Standard CAN Registers and Reset Values .....	234

SFR Definition 21.2. LIN0DAT: LIN0 Indirect Data Register .....	219
SFR Definition 21.3. LIN0CF: LIN0 Control Mode Register .....	220
SFR Definition 22.1. CAN0CFG: CAN Clock Configuration .....	236
SFR Definition 23.1. SMB0CF: SMBus Clock/Configuration .....	243
SFR Definition 23.2. SMB0CN: SMBus Control .....	245
SFR Definition 23.3. SMB0DAT: SMBus Data .....	247
SFR Definition 24.1. SCON0: Serial Port 0 Control .....	259
SFR Definition 24.2. SMOD0: Serial Port 0 Control .....	260
SFR Definition 24.3. SBUF0: Serial (UART0) Port Data Buffer .....	261
SFR Definition 24.4. SBCON0: UART0 Baud Rate Generator Control .....	261
SFR Definition 24.6. SBRLL0: UART0 Baud Rate Generator Reload Low Byte .....	262
SFR Definition 24.5. SBRLH0: UART0 Baud Rate Generator Reload High Byte .....	262
SFR Definition 25.1. SCON1: Serial Port 1 Control .....	267
SFR Definition 25.2. SBUF1: Serial (UART1) Port Data Buffer .....	268
SFR Definition 26.1. SPI0CFG: SPI0 Configuration .....	277
SFR Definition 26.2. SPI0CN: SPI0 Control .....	278
SFR Definition 26.3. SPI0CKR: SPI0 Clock Rate .....	279
SFR Definition 26.4. SPI0DAT: SPI0 Data .....	279
SFR Definition 27.1. CKCON: Clock Control .....	284
SFR Definition 27.2. TCON: Timer Control .....	289
SFR Definition 27.3. TMOD: Timer Mode .....	290
SFR Definition 27.4. TL0: Timer 0 Low Byte .....	291
SFR Definition 27.5. TL1: Timer 1 Low Byte .....	291
SFR Definition 27.6. TH0: Timer 0 High Byte .....	292
SFR Definition 27.7. TH1: Timer 1 High Byte .....	292
SFR Definition 27.8. TMR2CN: Timer 2 Control .....	296
SFR Definition 27.9. TMR2RLL: Timer 2 Reload Register Low Byte .....	297
SFR Definition 27.10. TMR2RLH: Timer 2 Reload Register High Byte .....	297
SFR Definition 27.11. TMR2L: Timer 2 Low Byte .....	298
SFR Definition 27.12. TMR2H: Timer 2 High Byte .....	298
SFR Definition 27.13. TMR3CN: Timer 3 Control .....	302
SFR Definition 27.14. TMR3RLL: Timer 3 Reload Register Low Byte .....	303
SFR Definition 27.15. TMR3RLH: Timer 3 Reload Register High Byte .....	303
SFR Definition 27.16. TMR3L: Timer 3 Low Byte .....	304
SFR Definition 27.17. TMR3H: Timer 3 High Byte .....	304
SFR Definition 27.18. TMRnCN: Timer 4 and 5 Control .....	308
SFR Definition 27.19. TMRnCF: Timer 4 and 5 Configuration .....	309
SFR Definition 27.20. TMRnCAPL: Timer 4 and 5 Capture Register Low Byte .....	310
SFR Definition 27.21. TMRnCAPH: Timer 4 and 5 Capture Register High Byte .....	310
SFR Definition 27.22. TMRnL: Timer 4 and 5 Low Byte .....	311
SFR Definition 27.23. TMRnH: Timer 4 and 5 High Byte .....	311
SFR Definition 28.1. PCA0CN: PCA0 Control .....	325
SFR Definition 28.2. PCA0MD: PCA0 Mode .....	326
SFR Definition 28.3. PCA0PWM: PCA0 PWM Configuration .....	327
SFR Definition 28.4. PCA0CPMn: PCA0 Capture/Compare Mode .....	328

## 4.3. QFN-40 Package Specifications

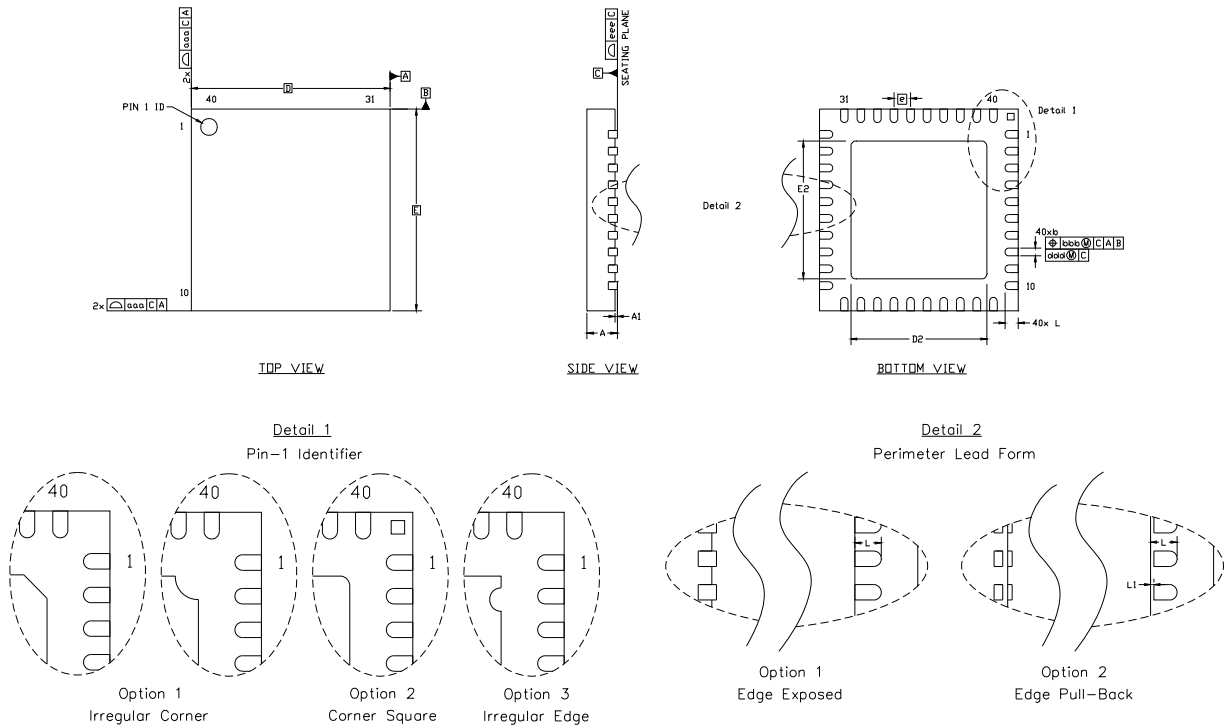


Figure 4.5. Typical QFN-40 Package Drawing

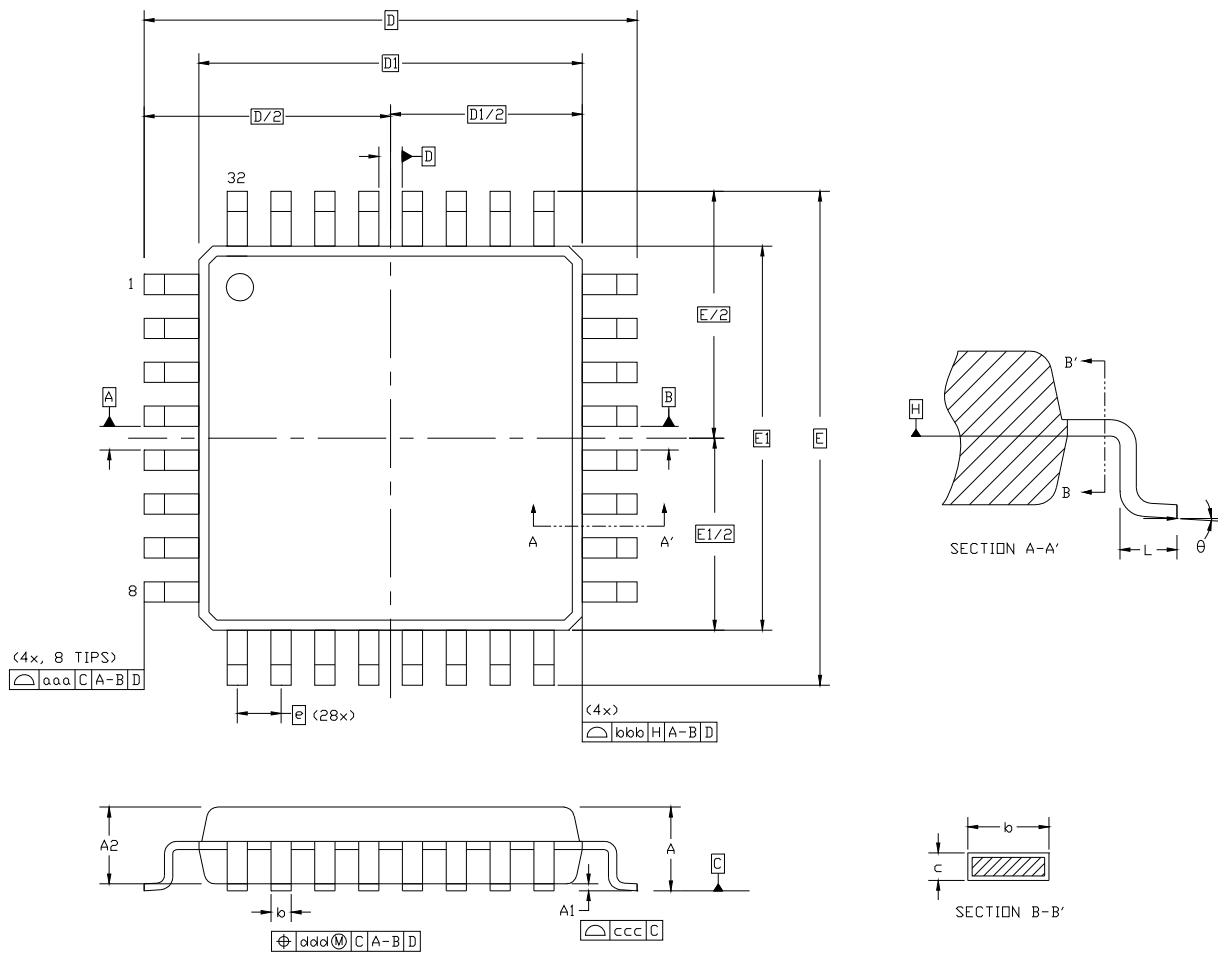
Table 4.5. QFN-40 Package Dimensions

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.80	0.85	0.90	E2	4.00	4.10	4.20
A1	0.00		0.05	L	0.35	0.40	0.45
b	0.18	0.23	0.28	L1			0.10
D	6.00 BSC			aaa			0.10
D2	4.00	4.10	4.20	bbb			0.10
e	0.50 BSC			ddd			0.05
E	6.00 BSC			eee			0.08

**Notes:**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation VJJD-5, except for features A, D2, and E2 which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

## 4.4. QFP-32 Package Specifications



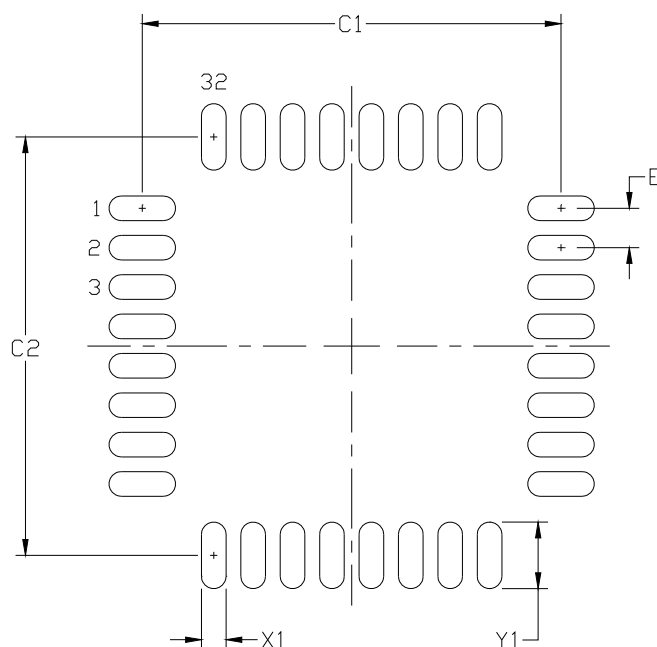
**Figure 4.7. QFP-32 Package Drawing**

**Table 4.7. QFP-32 Package Dimensions**

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	—	—	1.60	E	9.00 BSC.		
A1	0.05	—	0.15	E1	7.00 BSC.		
A2	1.35	1.40	1.45	L	0.45	0.60	0.75
b	0.30	0.37	0.45	aaa	0.20		
c	0.09	—	0.20	bbb	0.20		
D	9.00 BSC.			ccc	0.10		
D1	7.00 BSC.			ddd	0.20		
e	0.80 BSC.			θ	0°	3.5°	7°

**Notes:**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC outline MS-026, variation BBA.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.



**Figure 4.8. QFP-32 Package Drawing**

**Table 4.8. QFP-32 Landing Diagram Dimensions**

Dimension	Min	Max	Dimension	Min	Max
C1	8.40	8.50	X1	0.40	0.50
C2	8.40	8.50	Y1	1.25	1.35
E	0.80 BSC				

**Notes:**

**General**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

**Solder Mask Design**

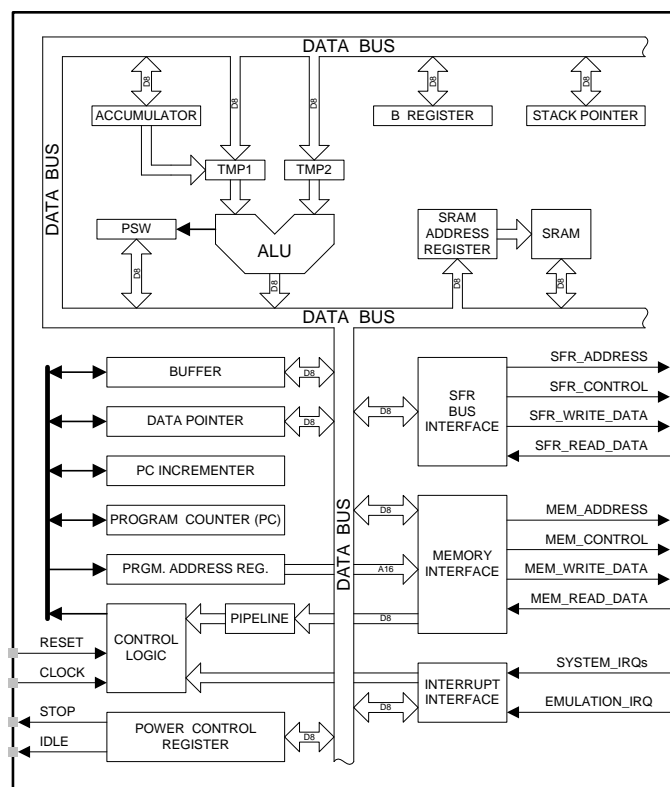
3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60µm minimum, all the way around the pad.

**Stencil Design**

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.

**Card Assembly**

7. A No-Clean, Type-3 solder paste is recommended.
8. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.



**Figure 11.1. CIP-51 Block Diagram**

With the CIP-51's maximum system clock at 50 MHz, it has a peak throughput of 50 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

## Programming and Debugging Support

In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2).

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources. C2 details can be found in "C2 Interface" on page 351.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.



## 15.4. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of  $V_{DD}$ , system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

### 15.4.1. $V_{DD}$ Maintenance and the $V_{DD}$ monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum  $V_{REGIN}$  rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external  $V_{DD}$  brownout circuit to the  $\overline{RST}$  pin of the device that holds the device in reset until  $V_{DD}$  reaches the minimum threshold and re-asserts  $\overline{RST}$  if  $V_{DD}$  drops below the minimum threshold.
3. Enable the on-chip  $V_{DD}$  monitor to the high setting and enable the  $V_{DD}$  monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the  $V_{DD}$  monitor and enabling the  $V_{DD}$  monitor as a reset source. Code examples showing this can be found in "AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the  $V_{DD}$  monitor and enable the  $V_{DD}$  monitor as a reset source inside the functions that write and erase Flash memory. The  $V_{DD}$  monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.

**Note:** The output of the internal voltage regulator is calibrated by the MCU immediately after any reset event. The output of the un-calibrated internal regulator could be below the high threshold setting of the VDD Monitor. If this is the case, and the MCU receives a non-power on reset (POR) when the VDD Monitor is set to the high threshold setting, the MCU will remain in reset until a POR occurs (i.e. VDD Monitor will keep the device in reset). A POR will force the VDD Monitor to the low threshold setting, which is guaranteed to be below the un-calibrated output of the internal regulator. The device will then exit reset and resume normal operation. It is for this reason Silicon Labs strongly recommends that the VDD Monitor is always left in the low threshold setting (i.e. default value upon POR). When programming the Flash in-system, the VDD Monitor must be set to the high threshold setting. To prevent this issue from happening and ensure the highest system reliability, firmware can change the  $V_{DD}$  Monitor high threshold, and the system must use an external supply monitor. For instructions on how to do this, see "Reprogramming the VDD Monitor High Threshold" on page 138.

5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

## SFR Definition 16.1. PCON: Power Control

Bit	7	6	5	4	3	2	1	0
Name	GF[5:0]						STOP	IDLE
Type	R/W						R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x87; SFR Page = All Pages

Bit	Name	Function
7:2	GF[5:0]	<b>General Purpose Flags 5–0.</b> These are general purpose flags for use under software control.
1	STOP	<b>Stop Mode Select.</b> Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0. 1: CPU goes into Stop mode (internal oscillator stopped).
0	IDLE	<b>IDLE: Idle Mode Select.</b> Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0. 1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)

---

**SFR Definition 18.1. EMI0CN: External Memory Interface Control**


---

Bit	7	6	5	4	3	2	1	0
Name	PGSEL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAA; SFR Page = 0x00

Bit	Name	Function
7:0	PGSEL[7:0]	<b>XRAM Page Select Bits.</b> The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. 0x00: 0x0000 to 0x00FF 0x01: 0x0100 to 0x01FF ... 0xFE: 0xFE00 to 0xFEFF 0xFF: 0xFF00 to 0xFFFF

**SFR Definition 20.24. P2SKIP: Port 2 Skip**

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD6; SFR Page = 0x0F

Bit	Name	Function
7:0	P2SKIP[7:0]	<b>Port 2 Crossbar Skip Enable Bits.</b> These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P2.n pin is not skipped by the Crossbar. 1: Corresponding P2.n pin is skipped by the Crossbar.

**SFR Definition 20.25. P3: Port 3**

Bit	7	6	5	4	3	2	1	0
Name	P3[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P3[7:0]	<b>Port 3 Data.</b> Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P3.n Port pin is logic LOW. 1: P3.n Port pin is logic HIGH.
<b>Note:</b> Port P3.1–P3.6 are only available on the 48-pin and 40-pin packages.				

# C8051F58x/F59x

## 21.7. LIN Registers

The following Special Function Registers (SFRs) and indirect registers are available for the LIN controller.

### 21.7.1. LIN Direct Access SFR Registers Definitions

#### SFR Definition 21.1. LIN0ADR: LIN0 Indirect Address Register

Bit	7	6	5	4	3	2	1	0
Name	LIN0ADR[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD3; SFR Page = 0x00

Bit	Name	Function
7:0	LIN0ADR[7:0]	<b>LIN Indirect Address Register Bits.</b> This register hold an 8-bit address used to indirectly access the LIN0 core registers. Table 21.4 lists the LIN0 core registers and their indirect addresses. Reads and writes to LIN0DAT will target the register indicated by the LIN0ADR bits.

#### SFR Definition 21.2. LIN0DAT: LIN0 Indirect Data Register

Bit	7	6	5	4	3	2	1	0
Name	LIN0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD2; SFR Page = 0x00

Bit	Name	Function
7:0	LIN0DAT[7:0]	<b>LIN Indirect Data Register Bits.</b> When this register is read, it will read the contents of the LIN0 core register pointed to by LIN0ADR. When this register is written, it will write the value to the LIN0 core register pointed to by LIN0ADR.

## 22. Controller Area Network (CAN0)

**Important Documentation Note:** The Bosch CAN Controller is integrated in the C8051F580/2/4/6/8-F590 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, refer to Bosch's C\_CAN User's Manual as an accompanying manual to the Silicon Labs' data sheet.

The C8051F580/2/4/6/8-F590 devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 22.1 shows an example typical configuration on a CAN bus.

Silicon Labs CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFRs) in the CIP-51.

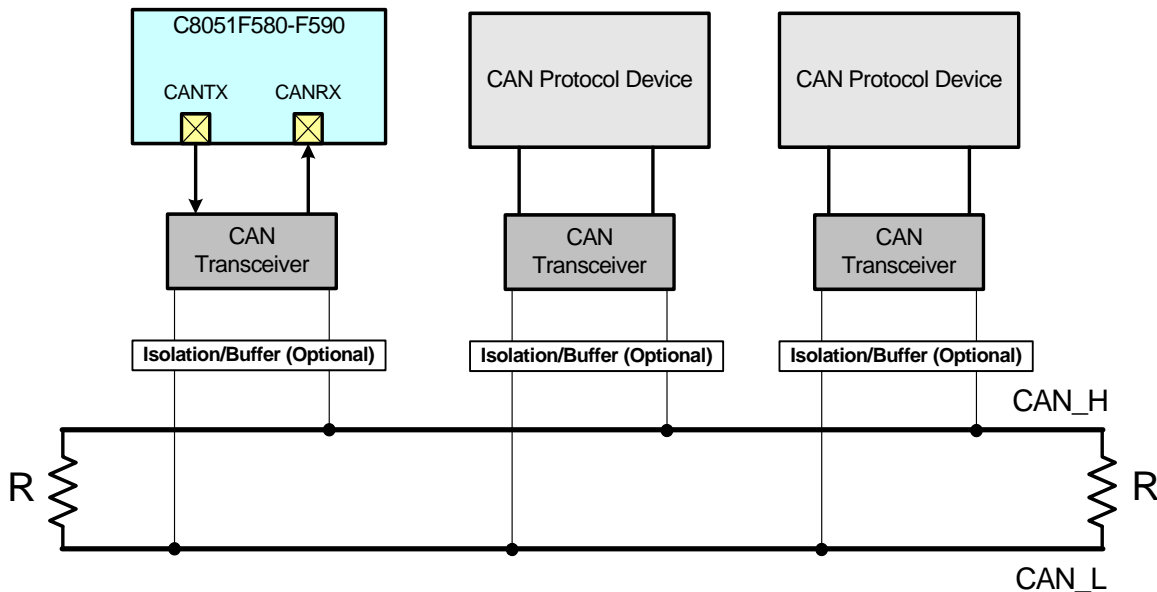


Figure 22.1. Typical CAN Bus Configuration

## SFR Definition 24.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	OVR0	PERR0	THRE0	REN0	TBX0	RBX0	TI0	RI0
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Address = 0x98; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	OVR0	<b>Receive FIFO Overrun Flag.</b> 0: Receive FIFO Overrun has not occurred 1: Receive FIFO Overrun has occurred; A received character has been discarded due to a full FIFO.
6	PERR0	<b>Parity Error Flag.</b> When parity is enabled, this bit indicates that a parity error has occurred. It is set to 1 when the parity of the oldest byte in the FIFO does not match the selected Parity Type. 0: Parity error has not occurred 1: Parity error has occurred. This bit must be cleared by software.
5	THRE0	<b>Transmit Holding Register Empty Flag.</b> Firmware should use or poll on TI0 rather than THRE0 for asynchronous UART writes that may have a random delay in between transactions. 0: Transmit Holding Register not Empty—do not write to SBUF0. 1: Transmit Holding Register Empty—it is safe to write to SBUF0.
4	REN0	<b>Receive Enable.</b> This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO. 0: UART1 reception disabled. 1: UART1 reception enabled.
3	TBX0	<b>Extra Transmission Bit.</b> The logic level of this bit will be assigned to the extra transmission bit when XBE0 is set to 1. This bit is not used when Parity is enabled.
2	RBX0	<b>Extra Receive Bit.</b> RBX0 is assigned the value of the extra bit when XBE1 is set to 1. If XBE1 is cleared to 0, RBX1 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled.
1	TI0	<b>Transmit Interrupt Flag.</b> Set to a 1 by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	<b>Receive Interrupt Flag.</b> Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 ISR. This bit must be cleared manually by software. Note that RI0 will remain set to '1' as long as there is data still in the UART FIFO. RI0 can be cleared after the last byte has been shifted from the FIFO to SBUF0.

## 26. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

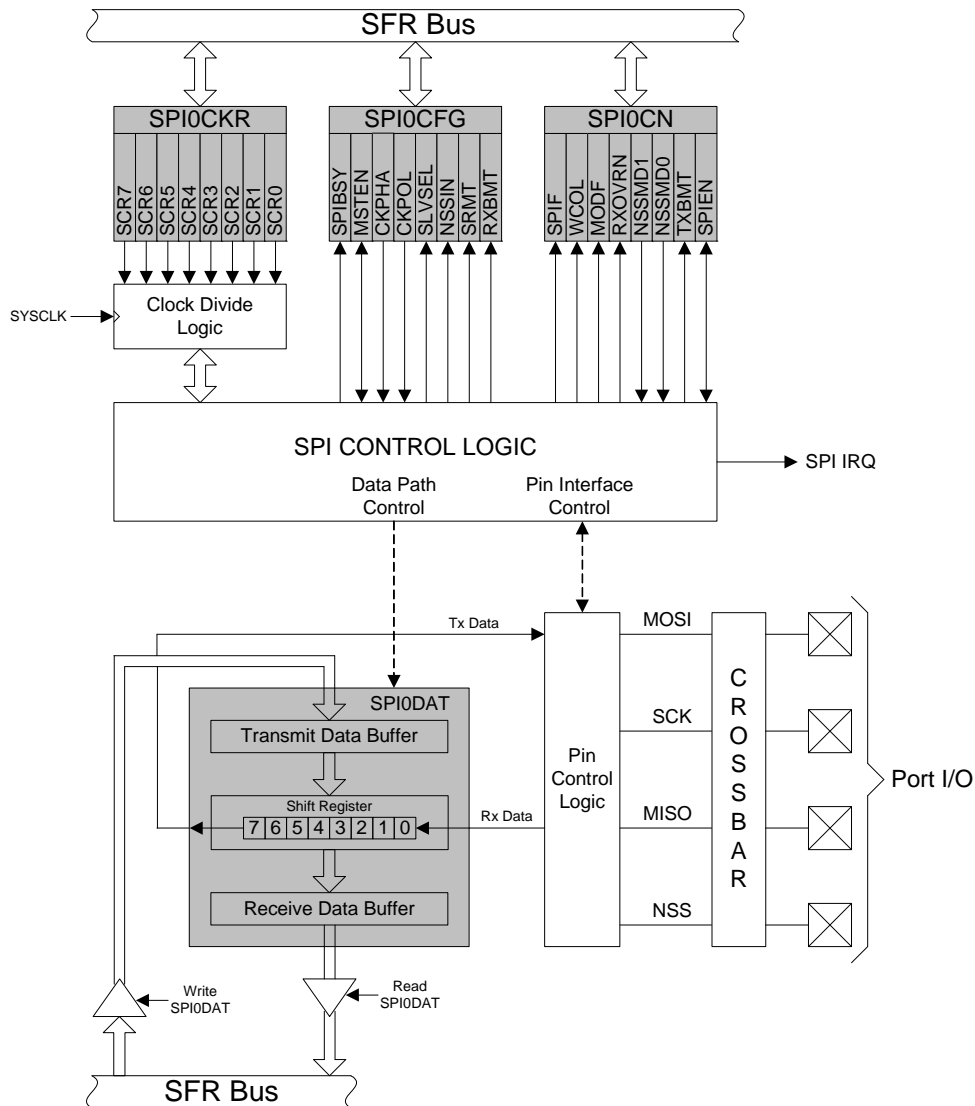
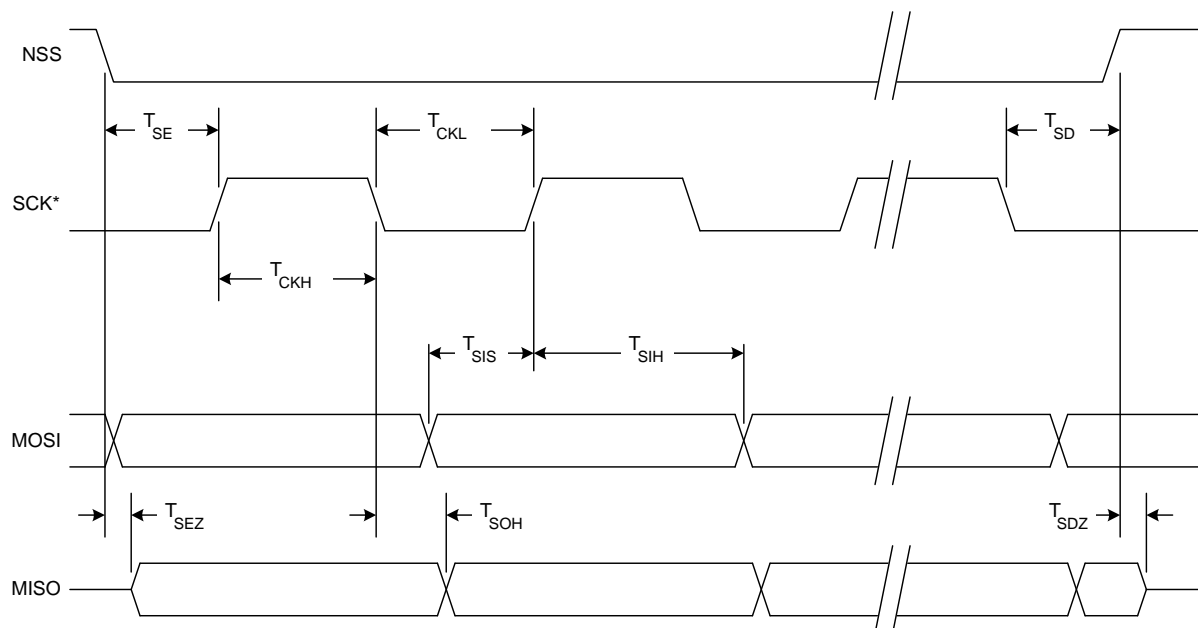


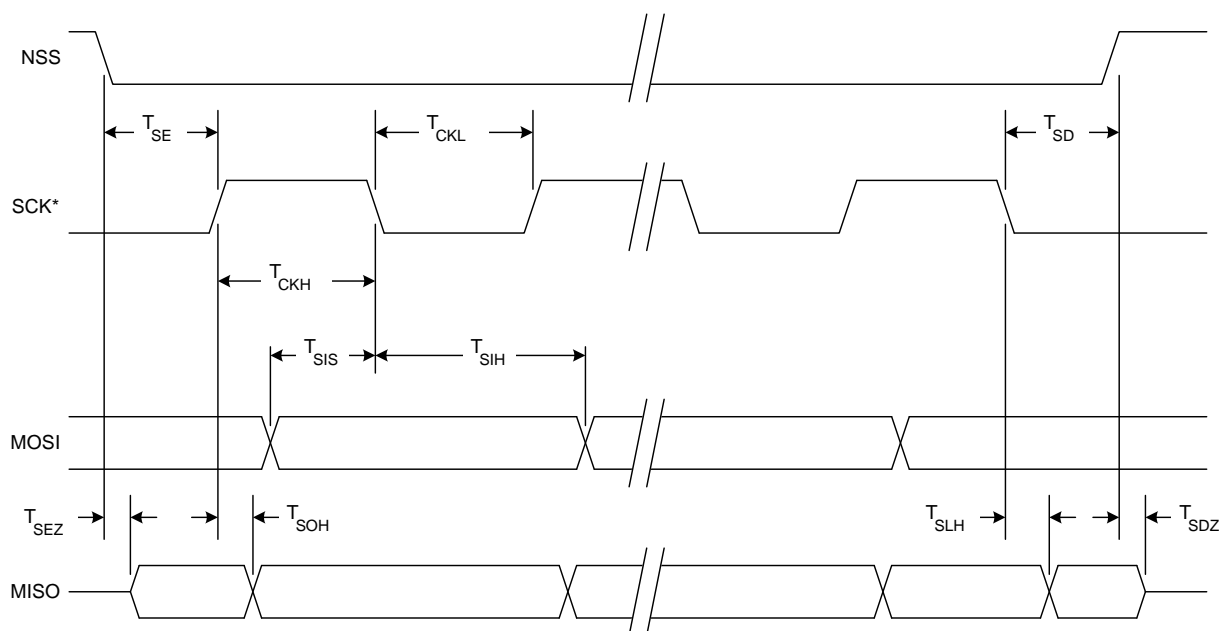
Figure 26.1. SPI Block Diagram





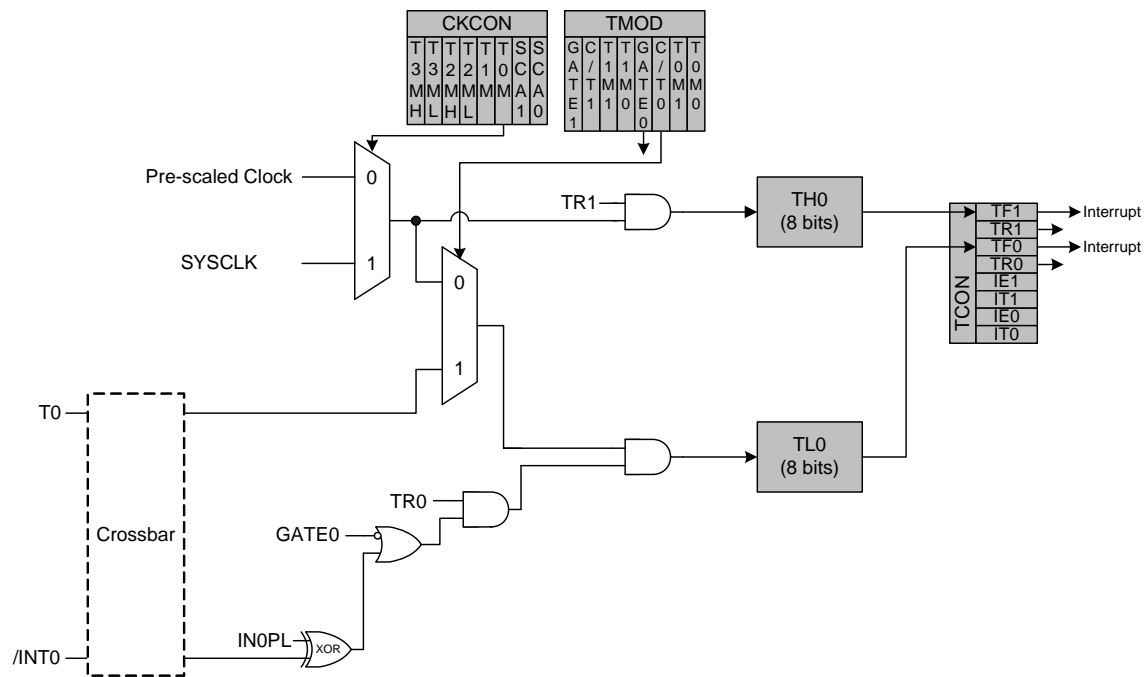
\* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 26.10. SPI Slave Timing (CKPHA = 0)**



\* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 26.11. SPI Slave Timing (CKPHA = 1)**



**Figure 27.3. T0 Mode 3 Block Diagram**

# C8051F58x/F59x

## SFR Definition 27.13. TMR3CN: Timer 3 Control

Bit	7	6	5	4	3	2	1	0
Name	TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3		T3XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x91; SFR Page = 0x00

Bit	Name	Function
7	TF3H	<b>Timer 3 High Byte Overflow Flag.</b> Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF3L	<b>Timer 3 Low Byte Overflow Flag.</b> Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. TF3L will be set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.
5	TF3LEN	<b>Timer 3 Low Byte Interrupt Enable.</b> When set to 1, this bit enables Timer 3 Low Byte interrupts. If Timer 3 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
4	TF3CEN	<b>Timer 3 Capture Mode Enable.</b> 0: Timer 3 Capture Mode is disabled. 1: Timer 3 Capture Mode is enabled.
3	T3SPLIT	<b>Timer 3 Split Mode Enable.</b> When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.
2	TR3	<b>Timer 3 Run Control.</b> Timer 3 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in split mode.
1	Unused	Read = 0b; Write = Don't Care
0	T3XCLK	<b>Timer 3 External Clock Select.</b> This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 3 clock is the system clock divided by 12. 1: Timer 3 clock is the external clock divided by 8 (synchronized with SYSCLK).

## 29.3. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8 to 11-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation. Table 29.2 summarizes the bit settings in the PCA1CPMn and PCA1PWM registers used to select the PCA1 capture/compare module's operating mode. All modules set to use 8, 9, 10, or 11-bit PWM mode must use the same cycle length (8-11 bits). Setting the ECCF1n bit in a PCA1CPMn register enables the module's CCFn interrupt.

**Table 29.2. PCA1CPM and PCA1PWM Bit Settings for PCA1 Capture/Compare Modules**

Operational Mode	PCA1CPMn								PCA1PWM				
Bit Number	7	6	5	4	3	2	1	0	7	6	5	4–2	1–0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX
<b>Notes:</b> <ol style="list-style-type: none"> <li>1. X = Don't Care (no functional difference for individual module if 1 or 0).</li> <li>2. A = Enable interrupts for this module (PCA1 interrupt triggered on CCFn set to 1).</li> <li>3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL1[1:0]).</li> <li>4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).</li> <li>5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA1CPHn and PCA1CPLn.</li> <li>6. E = When set, a match event will cause the CCFn flag for the associated channel to be set.</li> <li>7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.</li> </ol>													

## 29.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA1 clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 29.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA1CPHn}$$

Note: A value of 0x00 in the PCA1CPHn register is equal to 256 for this equation.

### Equation 29.1. Square Wave Frequency Output

Where  $F_{PCA}$  is the frequency of the clock selected by the CPS12–0 bits in the PCA1 mode register, PCA1MD. The lower byte of the capture/compare module is compared to the PCA1 counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA1CPLn. Frequency Output Mode is enabled by setting the ECOM1n, TOG1n, and PWM1n bits in the PCA1CPMn register. Note that the MAT1n bit should normally be set to 0 in this mode. If the MAT1n bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA1 counter and the 16-bit capture/compare register for the channel are equal.

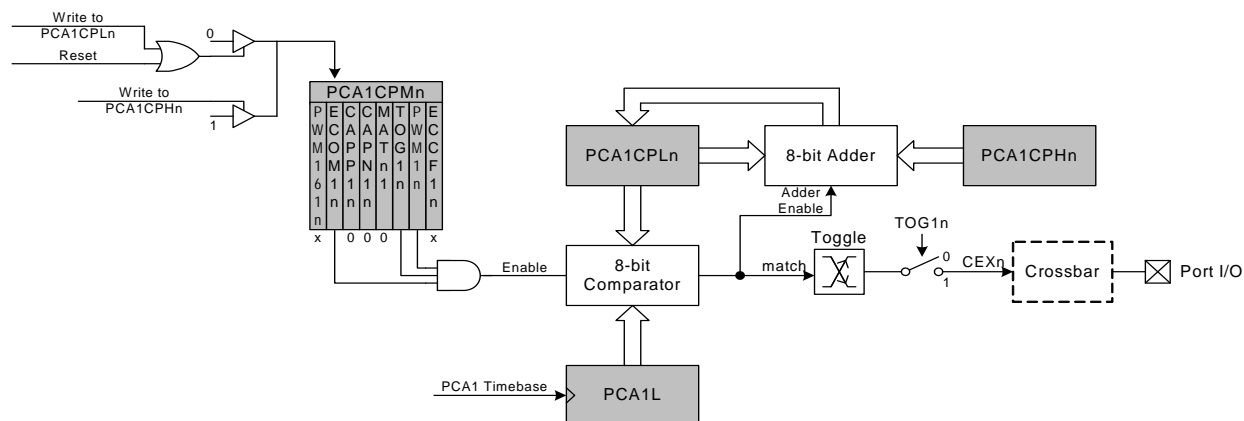


Figure 29.7. PCA1 Frequency Output Mode